



۱. هدفها

- آشنایی با پروتکل های پیام رسانی توزیع شده ^۱،
- آشنایی با مدیریت نشست ^۲،
- آشنایی با UNIX Socket API.

۲. مقدمه

می خواهیم یک برنامه برای ارتباط متنی چندین کاربر آماده کنیم. پروتکل های مختلفی نظیر IRC ^۳ وجود دارد که به صورت توزیع شده عمل می کنند. در این تمرین قصد داریم یک پروتکل پیشنهاد کنیم که از فرستادن همزمان یک پیام به چندین گره پشتیبانی کند. پروتکل پیشنهادی ما از یک درخت فراگیر جهت ارتباط استفاده می کند و از جهت توزیع شده بودن مانند IRC است.

۳. آشنایی با ساختار ارتباطی کارگزارها و کارخواهها

در ساختار درختی پیشنهادی (سدپ) هر گره می تواند یا کارخواه باشد یا کارگزار یا به صورت همزمان در هر دو نقش ظاهر شود. در این ساختار تمامی گره ها به جز ریشه ی درخت، تنها یک پدر دارند. در سدپ هر گره با یک

* با سپاس از علی فتاح المنان، عرفان عبدی، مهران خلدی، میلاد عسگری، محمد بخشعلی پور و پیمان عزتی

^۱Distributed messaging

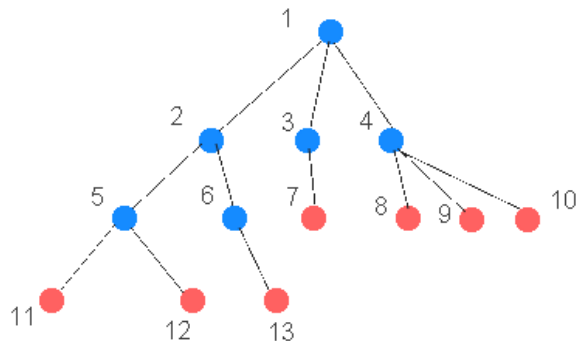
^۲Session management

^۳Internet Relay Chat

عدد یکتا مشخص می‌شود که این عدد، عددی بین یک تا ۱۰۰۰۰ است. گره‌ها هر شماره‌ای می‌توانند داشته باشند. هر گره باید از تمامی گره‌هایی که در زیردرخت آن قرار دارند، مطلع باشد. برای این منظور هرگاه گره‌ای یک پدر جدید انتخاب می‌کند، شماره‌ی خود و گره‌های زیردرخت خود را برای پدر می‌فرستد. پدر او نیز این اطلاعات را تا ریشه منقل می‌کند. هرگاه پدری اطلاعی از فرزند خود دریافت کرد، آن اطلاع و فرستنده‌ی آن را ذخیره می‌کند. این امر به منظور ارسال‌های آتی برای گره‌های زیردرخت انجام می‌شود. در ادامه توضیحات بیشتری در این زمینه آورده شده است.

همچنین اگر راسی پدر جدیدی انتخاب کرد، اطلاعات خود و گره‌های زیردرخت خود را برای پدر قدیم می‌فرستد تا نخست، پدر دانش خود را از زیردرخت به‌روز نماید و دوم این‌که با ارسال اطلاعات توسط پدر به ریشه، تمامی اجداد گره‌ی ذکر شده از جدا شدن این گره‌ها با خبر شوند. توجه داشته باشید، در شرایطی که یک راس قصد ترک شبکه را نیز داشته باشد، این اطلاعات را به پدر خود منتقل می‌کند تا اجداد این گره از حذف شدن زیردرخت و گره‌ی ذکر شده با خبر شوند. پس از این انتقال، گره می‌تواند از شبکه جدا شود.

هر گره می‌تواند برای یک گره یا گروهی از گره‌ها پیغام ارسال کند. به عنوان مثال فرض کنید در شکل ۱، گره‌ی ۱۱ بخواهد پیامی برای گره‌ی ۱۳ ارسال کند. برای این منظور، پس از ایجاد بسته‌ای به مقصد گره‌ی ۱۳، اگر گره‌ی ۱۳ در زیردرخت خود بود، بسته را برای فرزندی که درگاه ارتباطی با زیردرخت حاوی گره‌ی ۱۳ است، می‌فرستد. در غیر این صورت بسته را به پدر خود تحویل می‌دهد و پدر نیز عمل بالا را تکرار می‌کند تا بسته را به سمت مقصد هدایت کند.



شکل ۱: مثالی از ساختار شبکه

۴. پیاده سازی تمرین

در ابتدا که برنامه‌ی یک گره اجرا می‌شود، با وارد کردن یک عدد، شماره‌ی گره مشخص می‌شود. به عنوان مثال:

```
$/sdp.out 5
```

چهار نوع دستور زیر درون گره‌ها وارد می‌شود. توجه کنید که قبل از وارد کردن این دستورات، گره‌ی کارگزار بر روی پورت `i+7000` در حال گوش کردن است. منظور از `i` شماره گره‌ی کارگزار است. توضیحات بیشتر

در ادامه خواهد آمد:

۱.۴. مشخص کردن پدر

```
setparent <Name> <i>
```

این دستور برای مشخص کردن پدر در گره‌ها اعمال می‌شود. `i` شماره‌ی پدر و `Name` نام گره‌ی پدر است که با استفاده از `getaddrinfo`، `IP` پدر در دسترس است. این `IP` ممکن است `IPv4` یا `IPv6` باشد که پس از بررسی تمامی `IP` ها قابل دسترسی است. در صورتی که فرزند نتوانست به هیچ‌کدام از `IP` ها وصل شود، برنامه‌ی خود را پایان می‌دهد. توجه کنید پس از وارد شدن دستور `setparent` اطلاعات گره و زیردرخت آن برای گره `i` ارسال می‌شود. بنابراین نیاز است که برنامه‌ی مربوط به گره‌ی `i` در حال اجرا باشد. از طرف دیگر توجه کنید که با افزودن یک گره به شبکه، دوری ایجاد نشود. هنگامی که این دستور اجرا می‌شود، گره‌ی مورد نظر با یک ارتباط `TCP` به درگاه `i+7000` پدر خود متصل می‌شود. در واقع گره‌ی شماره `i` بر روی درگاه `i+7000` در حال گوش کردن است.

۲.۴. فرستادن پیام

دستور دوم برای ارسال پیام استفاده می‌شود و به شرح زیر است.

```
sendmessage d msg
```

که `d` شماره‌ی گره‌ی مقصد و `msg` یک پیام متنی است. با توجه به این که یک بایت برای طول بسته در نظر گرفته شده است، طول پیام متنی حداکثر ۲۵۵ کاراکتر می‌باشد. در تمام تست‌های ما این پیام متنی تنها شامل کاراکترهای انگلیسی بدون فاصله خواهد بود.

۳.۴. قطع شدن از شبکه

دستور سوم که برای قطع شدن از شبکه پیاده‌سازی می‌شود به صورت زیر است:

```
exit
```

با وارد شدن این دستور گره‌ی فرزند به پدر خود اطلاع می‌دهد که قصد ترک شبکه را دارد تا پدر اطلاعات خود را از زیردرخت خود، به‌روز کند و به اجداد خود انتقال دهد.

۴.۴. چاپ اطلاعات گره

دستور چهارم برای چاپ اطلاعات موجود در هر گره تدارک دیده شده است. این دستور به صورت زیر وارد می‌شود و خروجی زیر را نشان می‌دهد.

```
print
```

خروجی:

```
Parent: parent_id
```

```
C(c1): id1, id2, id3
```

```
C(c2): id4, id5
```

که در این نمایش، `parent_id` شماره‌ی گره‌ی پدر است و در صورت وجود نداشتن پدر برای گره‌ی مورد بررسی، این مقدار برابر `-1` است. `C(c1)` مجموعه‌ی شماره‌ی بچه‌های زیردرخت مربوط به فرزند اول است و شماره‌ی خود فرزند اول نیز در بین همین شماره‌ها قرار دارد و همین‌طور به ترتیب شماره‌های دیگر زیردرخت‌های مربوط به فرزند `i` ام چاپ می‌شوند. توجه کنید که این لیست و ترتیب چاپ اطلاعات زیردرخت فرزندان، مرتب شده و به صورت صعودی است.

به عنوان مثال با توجه به شکل ۱، برای گره‌ی شماره‌ی ۲ خروجی دستور `print` به صورت زیر است:

```
Parent: 1
```

```
C(5): 5, 11, 12
```

```
C(6): 6, 13
```

نکته: به فاصله‌ها و شکل خروجی دقت کنید.

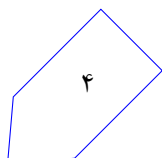
۵.۴. نقض پروتکل

در صورت نقض پروتکل و قواعد گفته شده توسط هر یک از گره‌های شبکه، گره‌ای که این نقض را متوجه شده است، باید ارتباط خود را با گره‌ی خطاکار قطع کند و سوکت خود را ببندد و به صورت عادی به کار خود ادامه دهد. بنابراین هم گره‌ی پدر و هم گره‌ی فرزند می‌توانند خطاکار تلقی شوند. توجه کنید که به روزرسانی اطلاعات اجداد این گره از طریق اطلاع‌رسانی گره‌ی پدر صورت می‌گیرد.

۶.۴. ساختار پیام‌ها

هر بسته‌ای که بین گره‌ها منتقل می‌شود، از چندین بخش تشکیل شده است که در زیر آن‌ها را شرح خواهیم داد. در تمامی بسته‌ها بایت اول مشخص‌کننده‌ی نوع بسته است.

با توجه به این‌که هدف ما این است که بتوانیم یک پیام را به تعدادی از گره‌ها به صورت همزمان برسانیم، ساختار زیر را در نظر گرفته‌ایم:



توضیحات	محتویات	بایت اول
این پیام نشان دهنده‌ی ترک یک گره از شبکه است. در صورتی که گره‌ی مورد نظر پدر داشته باشد، شماره‌ی خود و فرزندان خود را برای پدر خویش ارسال می‌کند تا آن‌ها را از لیست زیر درخت خود پاک کند. این نوع بسته تا هنگامی که به ریشه نرسد برای پدر ارسال خواهد شد. توجه کنید در صورت متوجه شدن نقض پروتکل توسط پدر نیز این پیام به اجداد ارسال می‌شود.	دو بایت اول محتویات تعداد گره‌های زیر درخت است. سپس دو بایت، دو بایت شماره‌ی اعضای زیر درخت قرار خواهد گرفت. (در لیست گره‌ها شماره‌ی خود گره، اولین شماره است).	۱۰
هرگاه <code>setparent</code> صدا زده شود (هنگامی که گره قصد متصل شدن به شبکه را داشته باشد)، لیست شماره‌ی خود و فرزندان خود را برای پدر جدید خویش ارسال می‌کند تا آن‌ها را به لیست زیر درخت خود اضافه کند. این نوع بسته تا هنگامی که به ریشه نرسد برای پدر ارسال خواهد شد.	دو بایت اول محتویات تعداد گره‌های زیر درخت است. سپس دو بایت، دو بایت شماره‌ی اعضای زیر درخت قرار خواهد گرفت. (در لیست گره‌ها شماره‌ی خود گره اولین شماره است).	۱۱
هنگامی که یک پیام درون گره‌ای ایجاد شود یا یک گره پیامی از فرزند خود دریافت کند، در صورتی که گره‌ی مقصد درون زیردرخت این گره نباشد، این نوع بسته را تولید می‌کند و برای پدر خویش می‌فرستد.	دو بایت اول محتویات شماره‌ی گره‌ی مقصد است. بایت بعدی طول پیام است و پس از آن پیام قرار خواهد گرفت.	۲۰
هنگامی که یک پیام درون گره‌ای ایجاد شود یا یک گره پیامی از فرزند خود دریافت کند، در صورتی که گره‌ی مقصد درون زیردرخت این گره باشد، این نوع بسته را تولید می‌کند و به فرزندی که گره‌ی مقصد را در زیردرخت خود دارد، ارسال می‌کند.	دو بایت اول محتویات شماره‌ی گره‌ی مقصد است. بایت بعدی طول پیام و پس از آن پیام قرار خواهد گرفت.	۲۱

۳۰	بایت اول طول پیام است و پس از آن پیام قرار خواهد گرفت.	هنگامی که یک بسته به مقصد می‌رسد، پس از چاپ محتویات پیام، این نوع بسته را تولید کرده و برای تمامی فرزندان خویش می‌فرستد تا آن‌ها نیز این بسته را به فرزندان خود ارسال کنند. هرگاه این بسته به گره‌ای می‌رسد، محتویات پیام بسته نمایش داده می‌شود. توجه کنید که هر گره پس از دریافت این بسته، محتویات پیام را در خط مجزای بعدی (New line) چاپ می‌کند.
----	--	--

به عنوان مثال فرض کنید، گرهی شماره‌ی ۴ برای گرهی شماره‌ی ۲ پیامی را ارسال کند. در این حالت گرهی شماره‌ی ۴ بسته‌ای با بایت اول ۲۰ می‌سازد و برای پدر خود (گره‌ی شماره‌ی ۱) ارسال می‌کند. این گره بسته‌ای با بایت اول ۲۱ ساخته و برای گرهی شماره‌ی ۲ ارسال می‌کند. گرهی شماره‌ی ۲ بسته‌ای با بایت اول ۳۰ می‌سازد و برای تمامی فرزندان خود ارسال می‌کند و فرزندان آن نیز این کار را تکرار می‌کنند.

۵. پیاده‌سازی

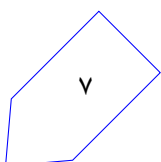
با توجه به اینکه ارزیابی و نمره‌دهی تمرین به صورت خودکار خواهد بود، توصیه می‌شود جهت ارتقای نمره‌ی خود مراحل زیر را برای پیاده‌سازی دنبال کنید.

۱. در سوکت‌های خود از گزینه `SO_REUSEADDR` استفاده کنید. این معیار برای گوش کردن سرور استفاده می‌شود و پیاده‌سازی خود و تست آن را آسان‌تر می‌کند. با تنظیم این گزینه مدت زمان وقفه‌ی سیستم عامل برای بازپس‌گیری پورت اشغالی توسط کارخواه حذف می‌شود و پورت به سرعت آزاد می‌شود.

۲. یکی از معیارهای ارزیابی، صحت اطلاعات درون گره‌ها است. به این معنی که در صورت رفتار صحیح گره‌ها، اطلاعات موجود در شبکه در زمان متناهی به‌روز می‌شود و تمامی اجزای شبکه اطلاعات درستی در خود خواهند داشت.

۳. اطلاعات موجود در مستند تمرین کارکرد یک کارخواه-کارگزار صحیح است. برنامه‌ی شما باید در برابر عملیات نامعتبر نیز کارا باشد. به عنوان مثال با وارد کردن دستورهای نامعتبر، برنامه‌ی شما باید از آن‌ها چشم‌پوشی کند و آن‌ها را در نظر نگیرد. از طرف دیگر هنگام دریافت بسته‌های نامعتبر، برنامه‌ی شما فرستنده‌ی بسته را به عنوان نقض‌کننده‌ی پروتکل شناسایی می‌کند و ارتباط خود را با آن قطع می‌کند. به عنوان مثال می‌توان ارسال بسته با بایت اول ۲۰ از پدر به فرزند را یک بسته‌ی نامعتبر دانست که نتیجه‌ی آن شناخته شدن

پدر به عنوان نقض کننده‌ی پروتکل توسط فرزند و تبدیل شدن شبکه کلّی به دو درخت مجزا است که گره‌ی فرزند در حالت قبل، ریشه‌ی درخت جدید خواهد بود. بنابراین برنامه‌ی شما در هر حالتی باید به اجرای خود ادامه دهد و تنها حالت خاتمه‌ی برنامه‌ی شما وارد کردن دستور `exit` است.



۶. نکات ضروری

- به علت اینکه نمره‌ی تمرین به صورت خودکار داده می‌شود، ساختار پیام‌های گفته شده باید تنها به صورت گفته شده باشد.
- نحوه‌ی ارزیابی ممکن است دچار تغییراتی شود و تست‌های دیگری اضافه شوند.
- در صورتی که هر مشکل یا پرسشی داشتید که فکر می‌کنید پاسخ آن برای همه مفید خواهد بود، آن را به گروه اینترنتی درس ارسال کنید.
- از فرستادن جواب تمرین به گروه اینترنتی درس خودداری کنید.
- تمام برنامه‌ی شما باید توسط خود شما نوشته شده باشد. فرستادن کل یا قسمتی از برنامه‌تان برای افراد دیگر، یا استفاده از کل یا قسمتی از برنامه‌ی فرد دیگری، حتی با ذکر منبع، تقلب محسوب می‌شود!
- پس از اتمام کارتان لازم است که فایل‌های خود را به همراه Makefile فشرده کرده و از طریق [وبسایت پرتو](#) ارسال نمایید.
- فایل ارسالی شما باید یک فایل zip باشد که پوشه‌ای در آن نیست و در آن فقط فایل‌های `.cpp` و `.h` شما به اضافه‌ی یک `Makefile` است که با اجرای دستور `make` فایل اجرایی `sdp.out` تولید می‌شود.